day31 factors
Due Monday 10/26/20

Today's program uses a bunch of skills and commands we have already learned to do some new things.

Today's program will:

- Ask the user for an integer using the try/except structure we learned last week
- State if the number is odd or even
- Find the sum of all the integers from 1 to the number
- Find all the factors of the number, store them in a list, then print the list. For example, the factors of 6 are: 1, 2, 3, and 6. The factors of 7 are: 1 and 7.
- Report if the number if prime or not. A number is prime if it can only be divided evenly by itself and 1 (for example, 7 is prime, 6 is not). An easy way to tell if your number is prime is to see how long the list of factors is: if it is 2 numbers long, your number is prime, otherwise, it is not prime.
- Ask if the user wants to run again.

Notes:

If you have forgotten how to do the try/except thing, go look it up. You did this last week.

As far as the odd/even check goes, this doesn't have anything to do with lists or for loops, you can just check if n % 2 == 0, that checks if the number is evenly divisible by 2 which would mean it is even.

To find the sum of the integers from 1 to the number, make a for loop where x starts at 1 and goes up to your number, and then add each x to a sum variable as you go up. To check your code, the sum of all integers up to and including 4 is 10 (1 + 2 + 3 + 4 = 10). You should create a variable *before the for loop starts* and put zero into it so that you can add up all the numbers. (`total = 0` outside of the loop, then inside the loop put `total = total + x`).

To find the factors of a number, create an empty list (factors = []) before your loop starts then do a for loop using a range() call from 1 to the number+1 in a variable x, and use the % operator with the number and each x. If there is no remainder (i.e. if n % x == 0), then x is a factor, and you should add it to a list variable using append(). Then you can print the list of factors out when the loop is done.

(continued on next page)

To find if a number if prime, check how many factors you have when you're done with the above loop. If you have two factors then the number is prime. For example, for 7 as n, you'd end up with a list containing [1,7], because no other integers divide evenly into 7, so just by the length of the list (the "len()" function) you can tell if a number is prime. (If the list has only 2 entries, your number is prime.)

Sample output:

```
Please enter an integer: 7
7 is odd.
The sum of all positive integers from 1 to 7 is 28
The factors of 7 are: [1, 7]
7 is prime.
Do another? (y/n)y

Please enter an integer: 240
240 is even.
The sum of all positive integers from 1 to 240 is 28920
The factors of 240 are: [1, 2, 3, 4, 5, 6, 8, 10, 12, 15, 16, 20, 24, 30, 40, 48,
60, 80, 120, 240]
Do another? (y/n)n
Too bad, I was having fun.
```